PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/00 | A1

(11) International Publication Number: **WO 98/42098**

(43) International Publication Date: 24 September 1998 (24.09.98)

(21) International Application Number: PCT/US98/04658

(22) International Filing Date: 11 March 1998 (11.03.98)

(30) Priority Data:
08/818,132 14 March 1997 (14.03.97) US

(71) Applicant: CRYPTOWORKS, INC. [US/US]; 2084 Union Street, San Francisco, CA 94123 (US).

(72) Inventor: LeBOURGEOIS, John, H.; 193 San Carlos Way, Novato, CA 94945 (US).

(74) Agent: WOLFELD, Warren, S.; Fliesler, Dubb, Meyer and Lovejoy LLP, Suite 400, Four Embarcadero Center, San Francisco, CA 94111-4156 (US).

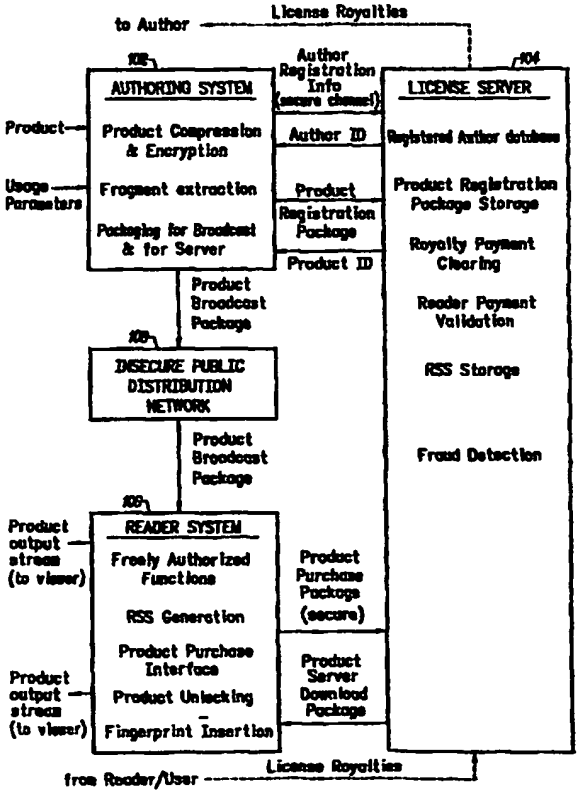
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: DIGITAL PRODUCT RIGHTS MANAGEMENT TECHNIQUE

(57) Abstract

A digital product is freely distributed through uncontrolled channels in encrypted form (108). Security fragment(s) of the encrypted product are withheld (102), and provided only upon communication with license server (104). The customer uses reader software (106) to purchase a license. Such software (106) examines components then present on the reader system to develop a reader system signature, which the license server (106) uses to encrypt the product decryption key and the security fragments before sending them to the reader system. When the customer wishes to use the product, a new reader system signature is generated and used to decrypt the product fragments.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

DIGITAL PRODUCT RIGHTS MANAGEMENT TECHNIQUE

BACKGROUND

1. Field of the Invention

The invention relates to controlled distribution of digital products in a manner that tends to ensure that authors receive proper royalty payments for their work.

2. Description of Related Art

Digitally encoded products, such as software, music, images and video segments are extremely important in today's economy. However, the ease and economy with which perfect copies can be made of digitally encoded products has promoted the widespread unauthorized copying and distribution of such products through such channels as user-to-user copying and sharing, digital data networks such as the Internet, and CD-ROM publishing. These distribution channels have made it difficult for the creators and copyright holders of such products to regulate the use of their products or to receive payment and registration information from their users. Consequently, digital product producers often forfeit substantial revenues and valuable information about their customer base and potential markets, while businesses and universities find themselves subject to legal prosecution and intimidation for copyright infringement.

But the problem of unauthorized distribution of digital products is not limited to the simple loss of revenues which rightfully belong to the original developer of the product, although that problem can be substantial by itself. There is also the additional problem that developers will be less willing to spend

substantial time and money in developing new digital products if they cannot depend on a market which pays fair value for their products.

In the past, when copying and distribution of products was more expensive, time consuming and imperfect, the creators of digital products could depend in part on economic factors as well as legal protections to minimize unauthorized copying and distribution. Neither is effective today with digital products, easy and inexpensive channels of distribution, and huge numbers of people participating in the illicit trade.

One technique that has been developed to deter unauthorized copying of digital products involves copy protection mechanisms built into hardware or software. However, copy protection mechanisms sometimes can inconvenience users who have a legitimate need for making copies. Copy protection mechanisms also can prevent digital product developers from taking advantage of the new distribution channels, such as the Internet.

Another method for controlling the distribution of digital products is described in Commerce Direct International (CDI), "Electronic Commerce", World Wide Web URL http://www.cdi.net/electron/commerce.htm (visited March 23, 1996) and CDI, "CDI In Action", World Wide Web URL http://www.cdi.net/cdiactio/cdinactn.htm (visited March 23, 1996), both incorporated herein by reference. The mechanisms described in these documents call for a customer to review an on-line catalog of products and choose one to buy. The customer's software then sends encrypted credit card information to the product distributor, and the product distributor transmits the product to the customer in a strongly

encrypted form. The customer's software then uses a "secret key" to decrypt the product and use it. Secure direct modem connections may be used instead of Internet connections at various parts of the process in order to further ensure that no one else can use the encrypted product.

The CDI technique suffers from at least two problems. First, while the encryption of a data product during transmission may be very strong, it is still theoretically possible for an unauthorized third party to decipher it. Second, even if the product remains secure during transmission, once it is decrypted by the customer, CDI's strong encryption techniques no longer protect the product against further unauthorized distribution. The above-cited documents indicate that content as well as executables can be "locked" to a specific registered machine while still allowing for unrestricted distribution of the software in a crippled or time-dated mode, but the documents do not explain how this is to be accomplished.

In Schull U.S. Patent No. 5,509,070, incorporated by reference herein, Schull describes a technique for distributing digital products (specifically software) by selling a password to the user. The user uses the password to unlock advanced features of the product, but the password is usable only on one machine. Thus, the technique allows users to freely distribute software to other machines and other potential users, but does not allow other users to take advantage of advanced features of the software without obtaining a new password which is specific to the new machine. In the Schull method, the user generates a "passwordable-ID" either from the user's voice, by reading the serial

number of the machine's CPU, or by any of a number of other suggested methods. The passwordable-ID is transmitted to a licensing system which uses the passwordable-ID to generate an encrypting seed for the software advanced features. The encrypted encrypting seed is then transmitted back to the user as a key for unlocking the advanced features of the software.

One drawback of the Schull method is that the password ultimately provided by the licensing system to the user is stored on the user's machine. Without certain extraordinary copy protection mechanisms applied to the password, the password could simply be copied to a new machine, thereby allowing a different user to use the advanced features of a pirated version of the product without paying a fair licensing fee. The extraordinary methods suggested by Schull to protect the password as stored on the user's machine, reimposes many of the same problems that formed the basis for finding non-objectionable copy protection mechanisms in the first place.

Several other technologies for preventing unauthorized distribution of digital products are set forth in the following World Wide Web pages: Northeast Consulting Resources, Inc., "Digital Rights Management Technologies", Web page URL http://www.ncri.com/articles/rights_management/ (October 1995: visited November 19, 1996); Intertrust Technologies Corporation, Web pages http://www.intertrust.com/architecture/overview.html, http://www.intertrust.com/architecture/flow.html, http://www.intertrust.com/architecture/stc.html, http://www.intertrust.com/products/applications.html, http://www.intertrust.com/products/sdk.html, http://www.intertrust.com/about/whoweare.html, http://www.intertrust.com/about/vision.html (all

visited November 19, 1996, and all incorporated herein by reference). However, all of the technologies described in these references suffer from one or all of the problems mentioned above, as well as others.

Accordingly, there is a deeply felt need for a new technology which will control the distribution of digital products via the Internet and other uncontrolled distribution channels such that a fair return to the originator of the product can be ensured without unduly hampering wide and free distribution of sufficient information about the digital product to enable customers to decide whether to purchase a license.

SUMMARY OF THE INVENTION

According to the invention, roughly described, a digital product is freely distributed through uncontrolled channels in encrypted form. Unencrypted preview material may also be provided in order to help the customer decide whether to purchase a license. In an aspect of the invention, one or more fragments of the encrypted product are withheld from uncontrolled distribution, and provided only upon communication with a license server. Unlike prior art mechanisms, which rely on practical limitations of computing power in order to prevent unauthorized product decryption, the technique of the present invention renders it literally impossible for an eavesdropper to recover the complete decrypted product without contacting the licensor. The product is not merely encrypted; to the extent of the security fragments, it is not even there.

In a second aspect of the invention, again roughly described, the customer purchases a license through the use of reader software which examines the

components then present on the reader system to develop a reader system signature (RSS). The reader system signature is transmitted to a license server which uses it either to encrypt the product decryption key, or to further encrypt the security fragments, or both, before sending them to the reader system. The reader system signature is not stored on the reader system. Rather, at the time that the customer would like to use the digital product (view an image or movie, listen to a sound or execute software, for example), a new reader system signature is generated in the same manner as the original reader system signature. The new reader system signature is used to decrypt the product decryption key and/or the double-encrypted security fragments, only at the time of use. If the user is attempting to use the digital product on a different machine, such as would be the case in the event of unauthorized distribution, then the new reader system signature will not work properly and the unauthorized user will not be able to use the digital product.

The above second aspect of the invention could be problematical in the event that the user upgrades or modifies the reader system. A mechanism that regenerates the reader system signature each time a digital product is to be used could prevent a legitimate user from using the product after such an upgrade. Accordingly, in another aspect of the invention, the machine authorization of the reader system allows a certain amount of "upgrade drift" before it is deemed advisable to check for unauthorized use.

In an embodiment, the reader system signature is determined by examining various components (hardware and/or software) of the reader system, to determine individual signatures for each component. The

individual component signatures are then combined to form the overall reader system signature, for example by a weighted sum of the individual component signatures or by concatenating the individual component signatures together. If the reader system signature is determined on the basis of a weighted sum (or equivalently, a weighted average) of the individual component signatures, then the amount of permissible upgrade drift can be expressed as a percentage; that is, if the reader system signature generated upon usage of the digital product differs from the reader system signature generated at the time the product is purchased by no more than a predetermined percentage or fraction, then the usage is considered authorized. If the reader system signature is generated as a concatenation of the individual component signatures, then the number of components which differ at usage time relative to purchase time can be specified not to exceed a specific count.

In a situation where the reader system signature generated at the time of purchase is not stored on the reader system, it can instead be uploaded to a license server. If the reader system signature generated at usage time is found by the reader system not to properly decrypt either the product decryption key or the product itself, then in an embodiment, the reader system can automatically contact the license server for reauthorization. The reader system uploads the newly generated reader system signature, and the license server performs the upgrade drift test in comparison with the reader system signature that was stored on the license server at the time of purchase. If the license server determines that the newly generated reader system signature is within the permissible upgrade

drift parameter, then it transmits a new product decryption key and/or new product security fragments, back to the reader system, encrypted according to the new reader system signature. The reader system is then able to decrypt the digital product and play, view or otherwise use it in the desired manner. (In order to assist in evaluating the permissibility of an upgrade drift, one embodiment of the reader system also uploads the raw component signatures of the computer system. This allows customer support to determine which components have changed.)

In an embodiment of the invention, the reader system signature is generated in dependence upon a component whose individual component signature carries with it external assurances of substantial uniqueness among all computer systems which could reasonably pose as authorized reader systems. Such a component signature can be used by itself, or in conjunction with other component signatures, in order to generate the overall reader system signature. If used by itself, then upgrade drift is usually not permissible without manual re-validation.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be described with respect to particular embodiments thereof, and reference will be made to the drawings, in which:

Fig. 1 is an overall symbolic diagram of a system according to the invention.

Fig. 2 is a block diagram illustrating the structure of a typical reader system of Fig. 1.

Fig. 3 illustrates the overall system flow for the authoring system of Fig. 1.

Fig. 4 is a flow chart illustrating the flow of a product registration segment of Fig. 3.

Fig. 5 is a flow chart illustrating the general operation of the reader system of Fig. 1.

Fig. 6 is a flow chart of the product purchase preparation step of Fig. 5.

Fig. 7 is a flow chart illustrating one technique for generating the reader system signature.

Figs. 8 and 9 together constitute a flow chart of steps which takes place in the license server 104 in response to receipt of a product purchase package.

Fig. 10 is a flow chart of the step in Fig. 8 in which the license server processes the customer's payment information.

Figs. 11-13 together constitute a flow chart of the step in Fig. 5 in which the reader system plays the digital product.

Fig. 14 is a flow chart illustrating the license server's operations in response to receipt of a re-validation package.

Figs. 15 and 16 are alternative details of the step in Fig. 14 in which the license server determines whether the difference between the two RSS's exceeds a threshold.

DETAILED DESCRIPTION

Fig. 1 is an overall symbolic diagram of a system according to the invention. The system has three primary components: an authoring system 102, a license server 104 and a reader system 106. In addition, the overall system is most useful when used with an uncontrolled distribution channel such as an insecure public distribution network 108 (e.g., the Internet). In general operation, the author or proprietor of one

or more digital products first uses the authoring system 102 to register with the license server 104 as an author. Author registration information is transmitted from the authoring system 102 to the license server 104, and an author ID is returned to the authoring system.

When the author has a digital product to market, the product is provided to the authoring system 102, together with certain usage parameters. The usage parameters include a set of free usage parameters and one or more sets of paid usage parameters. The authoring system compresses and encrypts the digital product (compression is optional), extracts one or more security fragments from the encrypted product and then packages the product for broadcast via the uncontrolled distribution network 108 and for upload to the license server 104. The authoring system then transmits the product registration package up to the license server 104 and receives a product ID in return. The authoring system also makes the product broadcast package available on the uncontrolled distribution network 108. Note that as used herein, a "product" can include one or more sub-products, all of which are considered herein to themselves be "products".

When a customer is interested in a particular digital product, he or she can download the product broadcast package from the uncontrolled distribution network 108. The customer utilizes the reading system 106 to perform those functions of the digital product which are freely authorized according to the free usage parameters that were previously specified by the author. Such functions can include, among other things, a preview of the digital product, and an indication of one or more licensing options which the

customer can purchase. If the customer chooses to purchase one of the license options, the reader system 106 examines certain components of the reader system and, in dependence thereon, generates a reader system signature (RSS). The reader system assembles a product purchase package including the RSS and payment information, and uploads it to the license server 104. The license server 104 processes the payment information and, if successful, transmits a product server download package back to the reader system. The reader system uses the product server download package to unlock the functions of the digital product which are authorized under the license option that the customer has purchased, and allows the user to use the product accordingly. In addition, the reader system 106 performs fingerprint and/or watermark insertion as described hereinafter.

The license server 104 performs a number of functions, including maintaining a database of registered authors and storing all of the product registration packages. The license server 104 also stores reader system signatures from customers, performs customer payment validation, as well as certain fraud detection functions as described below. The license server 104 also performs the functions of royalty payment clearing. Specifically, license royalties received from (or on behalf of) customers are properly accounted for and transferred to the proper authors after deduction of a commission.

In Fig. 1, the authoring system 102, the license server 104 and the reader system 106 are each illustrated as a respective individual block. Depending on the embodiment, each block might contain no more than a single computer, or in different

embodiments, different blocks can contain more than one computer. In one embodiment, one or more of the blocks 102, 104 and 106, for example the license server 104, contains a number of computers spread out over a great geographical area and interconnected by a network. The illustration of the authoring system 102, the license server 104, and the reader system 106 as single blocks is not intended to indicate that each must constitute only a single computer system or that each must be located at a respective single location.

Nor is there any requirement that computers used to form the authoring system 102, the license server 104, and the reader system 106 have any particular structure. Fig. 2 is a symbolic block diagram illustrating the structure of a typical computer system which may be used as an authoring system, a reader system or a license server. It comprises a CPU 202 and cache memory 204, both connected to a CPU bus 206. Interface circuitry 208 is also connected to the CPU bus 206. The interface circuitry 208 is further connected to a main memory 210, as well as to two I/O buses: PCI-bus 212 and ISA-bus 214. Connected to the PCI-bus 212 are sound and game controllers 216, a network adapter 232 and a display adapter 218, the last of which is further connected to a monitor 220. Connected to the ISA-bus 214 is a hard disk drive controller 222, a CD-ROM drive controller 224, a floppy disk drive controller 226, various I/O ports 228, and a boot PROM 230. Most of the peripheral components illustrated in Fig. 2 include on-board configuration data which can be read by the CPU 202. In addition, the boot PROM 230 includes a portion which is writeable by the CPU 202 to store configuration data. In general, the software to operate the authoring system

102, the license server 104 or the reader system 106 is stored on the disk drive controlled by the disk drive controller 222, and brought into main memory 210 as needed for execution. The computer system of Fig. 2 communicates with the other systems of Fig. 1, and with the distribution network 108, if appropriate, via the network adapter 232.

Fig. 3 illustrates the overall system flow for the authoring system 102. The authoring system flow is generally divided into two segments: an author registration segment 302 followed by one or more product registration segments 304. In the author registration segment 302, the author (or other proprietor) of one or more digital products enters his or her identification information. Such information can include, for example, the author's name, address, Social Security or other tax ID number, password or other challenge information (for confirmation of identity during customer service calls), e-mail address and/or telephone number (step 306). In a step 308, the authoring system uses this information to create an author registration package which is transmitted, in step 310, to the license server 104. The license server 104 adds the author and the author's identification information to its registered author database, and in step 312, the authoring system 102 receives and stores and author ID from the license server 104. The communication between the authoring system 102 and the license server 104 in the author registration segment 302 should take place via digital certificate and one-time secure channel, or by secure, signed electronic mail.

Fig. 4 is a flow chart illustrating the flow of a product registration segment 304 (Fig. 3). In a step

402, the author identifies one of possibly many digital products to the authoring system 102 and enters usage parameters. The digital product is identified, for example, by identifying a filename within which the digital product is stored. The usage parameters can include such parameters as the number of copies which will be permitted to be made on the reader system, whether the reader system will be authorized to save the digital product to a hard disk, whether printing will be enabled, whether preview is enabled, and the amount of RSS drift which will be permitted on a reader system before manual reauthorization will be required. The usage parameters may be specified as several options, including a set of free usage parameters (for which no payment is required) and one or more options of purchasable usage parameters (functions requiring a purchased license). In an embodiment, the author can also indicate at this point whether the product should be compressed.

In a step 404, if preview is to be enabled, the reader system extracts the appropriate preview material with the assistance of the author. The entire digital product is then encrypted in a step 406. In the product encryption step 406, the product is first compressed (step 408) by any known algorithm. For example, the product can be compressed using a Lempel-Ziv algorithm or by a Huffman encoding algorithm. The compression step 408 is considered part of the product encryption step 406 because compression is, in effect, a form of encryption; it is very difficult to recover the original uncompressed product unless the algorithm used for compression is known. Compression is optional because for certain kinds of products, the benefits to be gained by compression are outweighed by the

performance degradation that compression/decompression often entails.

In a step 410, a product encryption key is generated. The key can be generated in any known manner; for example, by a pseudo-random number generator using a seed derived from the time period between two successive user key strokes. In step 412, the compressed digital product is encrypted using the encryption key developed in step 410. Again, any known key-based symmetric encryption algorithm can be used (as long as the correct complementary algorithm is used for decryption on the reader system 106). One such well-known encryption algorithm is DES, described in National Institutes of Standards and Technology, "Data Encryption Standard," FIPS Publication No. 46-1 (January 1988), incorporated by reference herein. Another is Triple DES (also known as DES-3), and yet another is RC-5. RC-5 is described in R. W. Baldwin and R. Rivest, "The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms", INTERNET-DRAFT (March 1996), available from ftp://ftp.nordu.net/internet-drafts/draft-baldwin-rc5-00.txt, visited March 4, 1997, incorporated herein by reference.

The result of product encryption step 406 is referred to herein as an encrypted "version" of the digital product. As used herein, a "version" of a digital product is still considered to be the digital product, because it continues to include all the information of the digital product. A native "version" of a product also is nevertheless "the product". A digital product can exist in several versions, each of which is a reversibly processed version of the native version.

After the product encryption step 406, in step 414, the authoring system 102 generates a digest of the encrypted product. Any suitable digesting algorithm can be used for this purpose including, for example, an error-correcting code (ECC) generator or the well-known SHA-1 algorithm. The SHA-1 digesting algorithm is described National Institute of Standards and Technology (NIST), FIPS Publication 180: Secure Hash Standard (SHS) (May 1993), as amended by National Institute of Standards and Technology (NIST) Announcement of Weakness in the Secure Hash Standard (May 1994), both incorporated herein by reference.

In a step 416, the authoring system 102 separates one or more "security fragments" in the encrypting product. The security fragments preferably constitute only a small portion of the overall digital products; for example, 1-5% of the byte count of the overall product not to exceed, for example, 30K bytes. The number and size of the security fragments can be predetermined and constant for all digital products, or it can be dynamically selected or optimized for different digital products. It is desirable to minimize the size of the security fragments while maximizing the effect that their absence would have in helping to prevent unauthorized usage of the digital products. One way of maximizing such effect is to include the beginning of the digital product in one of the security fragments. For many encryption algorithms, as well as data compression algorithms, it is much more difficult to decrypt (decompress) the portions that remain, if the beginning portion is missing. Also, if the digital product includes a header at the beginning, absence of the beginning portion of the product often makes it difficult to use any of the remaining portion

of the product even if such remaining portion can be decrypted.

In a step 418, the authoring system 102 generates a digest of the encrypted product less the security fragments. Again any digesting algorithm, such as SHA-1, can be used in this step. In step 420, the authoring system 102 creates a product registration package and transmits it to the license server in step 422. The license registration package can form part of a digital certificate in one embodiment. The license server stores the product registration package and returns a product ID to the authoring system 102 (step 424). The reason that the security fragments should be kept as small as possible is to minimize the storage capacity requirements of the license server. In a step 426, the authoring system 102 creates a product broadcast package for the digital product and makes it available (step 428) via any uncontrolled distribution network, such as the Internet.

The product registration package, which can also be digitally certified, includes the following items:

- author ID
- usage parameters (both the free usage parameters and the usage parameters at various purchasable options)
- the encrypted security fragments
- a product decryption key (complementary to the product encryption key of step 410)
- the digest of the encrypted product less security fragments
- digest of the full encrypted product

The product decryption key referred to above is whatever key is required to decrypt the results of the encryption process of step 412. For symmetric

encryption algorithms, such as for DES or RC-5, the product decryption key will be the same as the key used for product encryption in step 410; for asymmetric encryption algorithms (such as for public key encryption), the product decryption key might be different than the product encryption key (such as for RSA encryption). In both cases, the product decryption key is referred to herein as being "complementary" to the key used for product encryption in step 410.

The product decryption key is the only segment of the product registration package which should be transmitted to the license server securely. Other segments can be transmitted securely if desired or convenient, but this is not necessary. Security of the product decryption key can be ensured, for example, by public/private key encryption of the product decryption key itself, or by connecting to the license server 104 via a secure network.

The product broadcast package contains the following items. If more than one sub-product is included in the product broadcast package, then the package contains a set of these items for each sub-product.

- product ID
- preview material (unencrypted)
- free usage parameters
- purchasable usage parameter options
- Universal Resource Locator (URL) of license server 104
- copyright notice
- contact information for assistance or questions
- encrypted product less security fragments

As can be seen, whereas the great bulk of the product is transmitted via the uncontrolled distribution network 108, not only is it encrypted, but it is also incomplete so that even if it could be decrypted, for example by a powerful computer system, important fragments of the product simply are not there.

Fig. 5 is a flow chart illustrating the general operation of the reader system 106. When a user installs the reader software on the reader system 106, the reader software automatically generates an installation ID. The installation ID, which is stored on disk in the reader system 106, is a substantially unique identifier of the installation. The installation ID is stored in such a way that if the particular installation of the reader system software was to be copied to a different system, the installation ID would likely be copied as well.

In a step 502, the customer/user of the reader system 106 downloads the product broadcast package for a digital product that he or she wishes to examine. In a step 504, the customer performs one or more of the freely authorized functions, including a preview of the material in the digital product (if available). For example, if the digital product is a sound, the preview material might be a portion of that sound. If a digital product includes a movie, then the preview might be a portion of the movie, or a trailer. If the digital product is an image, then the preview material might include a thumb nail of the image. If the digital product is text, then the preview material might include an abstract.

In a step 506, the customer chooses to buy a license in accordance with one of the license options

made available in the usage parameters. In step 508, the customer enters his or her identification information, payment and contact information. The identification information can include the customer's name, address and phone, or optionally an arbitrary privacy ID. Such identification is desirable so that the licensing authority can identify the customer if and when the customer calls in for support. The payment information can include a credit card number and authorization and/or digital money information. Various forms of digital money are described in Daniel C. Lynch, "Digital Money" (John Wiley & Sons, 1996), incorporated herein by reference.

In a step 510, the reader system 106 prepares a product purchase package described hereinafter, and in a step 512, the reader system 106 uploads the product purchase package to the license server 104 at the URL identified in the product broadcast package. Note that multiple or bounce URLs may be available for this purpose. The product purchase package may be transmitted by secure channel and/or encrypted in a digital certificate. Assuming the product purchase package is in order, then in a step 514, the reader system receives the product server download package for the desired digital product. The product server download package need not be transmitted by secure channel. In a step 516, the reader system stores the product broadcast package on the reader system (or at least accessibly to the reader system) in conjunction with the product server download package. Either at that time or at a later time, at the customer's request, the reader system plays (or otherwise uses) the product (step 518). All further use of the product

by the customer occurs on the reader system 106 through the reader system software.

Fig. 6 is a flow chart of the step 510 (Fig. 5) in which the reader system 106 prepares the product purchase package. In a step 602, the reader system 106 generates a reader system signature (RSS) for the reading system 106. The manner in which the RSS is generated is described hereinafter. In a step 604, the reader system 106 retrieves the previously generated installation ID, and in a step 606, the reader system generates a digest of the encrypted product less the security fragments (as obtained from the product broadcast package). The digest performed in step 606 should be the same as that performed by the authoring system 102 in step 418 (Fig. 4).

In a step 608, the reader system 106 extracts the product ID from the product broadcast package and in a step 610, the reader system assembles the product purchase package.

The product purchase package includes the following items:

- product ID
- customer's installation ID
- customer's identification information (or privacy ID)
- customer's payment information
- customer's contact information (including information on where to send the product server download package)
- RSS of the reader system 106
- generated digest of the encrypted product less security fragments

The reader system signature can be generated in a number of different ways in different embodiments. In one embodiment, the system takes advantage of serial numbers or other identifying data which may be present in the reader computer system, and which carry external assurances of substantial uniqueness. That is, many computers when manufactured are assigned a serial number or other indicator which the manufacturer of the computer, or some other authority, guarantees to be unique. For example, Apple MacIntosh computers, when manufactured, are assigned an Ethernet address which is unique to that specific computer. Alternatively, the identifier can be assigned in software, such as in the operating system of the computer. It is not essential that whatever authority assigns the serial number guarantee uniqueness; it is sufficient only in that it be extremely unlikely that two computer systems which can act as reader systems 106 carry the same identifier. This is the case where, for example, the number carries external assurances of substantial uniqueness, such as in the case of Ethernet addresses.

In another embodiment, the reader system signature does not rely on a component having an identifier that carries external assurances of substantial uniqueness. Instead, a plurality of components (hardware or software) are examined to determine individual component signatures. The individual component signatures are then combined to form the overall reader system signature. In one embodiment, the individual component signatures are all concatenated together in a predetermined sequence to form the overall reader system signature. The individual component signatures may be digested prior to concatenation in order to limit their size to the

predefined field size. In another embodiment, optionally after digesting, the individual component signatures are averaged or summed together to form the overall reader system signature. The individual component signatures can be weighted prior to combination, in order to reduce the impact on the reader system signature that would result from changes in components that are more frequently subject to upgrade or replacement.

In one embodiment, the reader system 106 generates the reader system signature in dependence upon component signatures from the following components, to the extent present in the system. Except as indicated below, most of the component signatures set forth in this list are readable either from the CMOS or from a device manager driver. This is only an illustrative list; other embodiments can refer to other components not on this list.

Hard Disk Drive

- drive ID
- numbers of cylinders, sectors and heads
- drive defective sector map (obtained from sector 0)
- drive name
- drive manufacturer

Floppy Disk Controller

- I/O addresses and settings
- interrupt assignments
- manufacturer name

Monitor

- monitor name
- monitor type

Display Adaptor

- device name
- on-board memory

Mother Board

- CPU type
- CPU speed
- total memory present
- total cache present
- cache timings (measured empirically)

Ports

- I/O addresses and settings
- interrupt assignments

Sound, Video and Game Controllers

- device name
- driver name
- driver version

System Devices

- CMOS profile

In yet another embodiment, a combination of individual component signatures also includes one or more component signatures that carry external

assurances of substantial uniqueness, to the extent such a component exists in the machine.

Fig. 7 is a flow chart illustrating one technique for generating the reader system signature for reader system 106. In a step 704, it is determined whether the reader system 106 includes a component which has an ID that carries external assurances of substantial uniqueness. If so, then the reader system signature is given by the component ID of that component (step 706). If not, then in step 708, the reader system 106 obtains the data regarding the above-listed components to the extent present in the reader system 106. In a step 710, each of the individual component signatures is digested, and they are combined in step 712 to form the reader system signature.

Returning to Fig. 5, as previously mentioned, after the reader system prepares and uploads a product purchase package to the license server (step 512), the license server 104, if everything is in order, returns a product server download package to the reader system 106 (step 514). Fig. 8 is a flow chart of the steps which takes place in the license server 104 in response to receipt of a product purchase package. As used herein, steps which take place "in response to" a predecessor event, do so if the predecessor event influenced the performance of such steps. If there is an intervening time period, the performance of the steps can still be considered "responsive" to the predecessor event. If the performance of the steps depends on more than one predecessor event, then the steps are considered performed in response to *each* of the predecessor events.

In a step 802, the license server 104 compares the digest from the product purchase package with the

digest of the encrypted product less security fragments as stored on the server 104 with the product registration package for the product ID referred to in the newly received product purchase package. If the two digests do not match, then it is very likely that either the product broadcast package has been tampered with prior to generation of the digest on the reader system 106, or the customer is attempting to obtain the product server download package fraudulently. In this case, the license server 104 returns a package to the reader system 106 indicating that the attempt to purchase a license was unsuccessful (step 804).

If the two digests do match, then in step 806, the license server 104 processes the customer's payment information. If there is an error in this process, then again the license server returns an indication to the reader system 106 that the customer's attempt to purchase a license has been unsuccessful (step 804).

Assuming the customer's payment information was processed successfully, in step 808, the license server 104 stores the customer's RSS obtained from the product purchase package in conjunction with the customer's installation ID, also obtained from the product purchase package. This information need not be stored "on" the license server 104 itself, as long as it is stored in a manner in which it is "accessible" to the license server 104 at a future time.

In a step 810, the license server 104 performs certain fraud detection checks. This step is optional in different embodiments, as indicated by the dotted line surrounding the box in Fig. 8. If performed, the fraud detection step 810 can include a check of the installation ID from the product purchase package

against the installation IDs that have been stored previously on the license server for other product purchases. If a large number of purchases have been made using product purchase packages specifying the same installation ID, then it is likely that someone has altered an installation of the reader system software and is passing it around to different customers who are using it to purchase licenses. The same is true if the same license has been purchased several times from the same installation ID, or if several significantly varying reader system signatures have been stored in the license server's database in conjunction with the same installation ID. A number of other fraud detection mechanisms can also be employed. In any event, an investigation is warranted if step 810 suggests that an altered version of the reader system software might be being distributed.

The flow chart of Fig. 8 continues after step 810 with step 902 in Fig. 9, as indicated by the symbol "9" in both Figs. 8 and 9.

In Fig. 9, in step 902, the license server 104 further encrypts the already once-encrypted security fragments (from the product registration package) using the customer's RSS as a key. The key used in step 902 need not be the RSS exactly; it can be some other number which depends on the RSS. For example, it can be a digest reduction of the RSS from the customer's product purchase package. In any event, step 902 results in "double-encrypted" security fragments from the digital product.

In step 902, the product decryption key from the product registration package is also encrypted using the customer's RSS (or a number derived therefrom) as a key. Note that in a different embodiment, either

step 902 or step 904 can be omitted, although such an omission would likely reduce the security of the overall system.

In a step 906, the license server 104 assembles the product server download package, and in a step 908, it transmits the product server download package back to the reader system 106. Processing then resumes in the reader system 106 at step 514 (Fig. 5).

The product server download package includes the following items:

- product ID
- paid usage parameters
- payment confirmation information (such as a credit card payment confirmation)
- digest of full encrypted product (from the product registration package stored on the license server 104)
- encrypted product decryption key (product decryption key encrypted with the customer's RSS)
- double-encrypted security fragments (encrypted security fragments further encrypted with the customer's RSS)

Fig. 10 is a flow chart of the step 806 (Fig. 8), in which the license server 104 processes the customer's payment information. Fig. 10 illustrates the process where the customer is paying by credit card; a similar process would take place where the customer is paying with digital money or in some other payment form.

In step 1002, the license server 104 transmits the charge information to a credit card clearing house. The credit card clearing house returns either an approval code or an error. In step 1004, if an error was received, then the license server 104 returns an

error to step 806 (Fig. 8) (Step 1006). If an approval code was received, then in step 1008, the license server 104 credits the author's account with the amount of the approved purchase price less a commission. In step 1010, the license server 104 returns successfully to the step 806 (Fig. 8).

Returning to Fig. 5, as previously mentioned, each time the customer desires to use the digital product, he or she does so using the reader system software on the reader system 106. Fig. 11 is a flow chart of the step 518 in which the reader system plays the digital product. (The terms "play", "view" and "use" are used interchangeably herein as regards a digital product.) Referring to Fig. 11, in a step 1102, the reader system 106 regenerates the RSS for the reader system. This step takes place using the same algorithm that was used in step 602 (Fig. 6) when the RSS was generated for preparation of the product purchase package. In a step 804, the reader system 106 decrypts the double-encrypted security fragments using the new RSS as a key. As mentioned with respect to step 904 (Fig. 9), the key used in step 1104 need not be the RSS identically; another number which depends on the RSS can be used instead. However, whatever algorithm is used to derive the key from the RSS in step 1104 should be the same as that used in step 904.

In step 1106, the reader system 106 merges the encrypted security fragments into the encrypted product less the encrypted security fragments, thereby assembling a complete, but still encrypted, version of the digital product. In step 1108, the full encrypted digital product is digested using the same algorithm as was used originally by the authoring system 102 in step 414 (Fig. 4). In step 1110, the reader system 106

determines whether the newly calculated digest matches the digest which was provided by the license server 104 in the product server download package. If so, then usage of the product on the reader system 106 is authorized. Flow continues with the flowchart of Fig. 12 (as indicated by the number "12" in the small circles in both Figs. 11 and 12). If the two digests do not match in step 1110, then usage of the product on the reader system 106 is not immediately authorized. Flow continues with the flowchart of Fig. 13, as indicated by the number "13" in the small circles in both Figs. 11 and 13.

Referring to Fig. 12, since the two digests match in step 1110, the current reader system signature has been confirmed to be the same as that which was used when the reader system 106 first uploaded its product purchase package in step 512 (Fig. 5). It is also the same RSS that was used to encrypt the product decryption key as downloaded from the license server 104 to the reader system 106 in steps 908 and 514. Accordingly, in step 1202, the reader system 106 decrypts the product decryption key from the product server download package using the current RSS. Again, the actual key used to decrypt the product decryption key in step 1202 need not be identical to the current RSS, as long as it is dependent thereon, and as long as the algorithm to generate the key is the same as that which was used to generate the key with which the product decryption key was originally encrypted in step 904 (Fig. 9).

In step 1204, the reader system 106 decrypts the merged encrypted product using the product decryption key that was decrypted in step 1202. In step 1206, if the decrypted product was compressed, it is now

decompressed using an algorithm complementary to that used by the authoring system in step 408 (Fig. 4). The resulting decompressed digital product is transmitted in step 1210 to an appropriate viewer.

It will be appreciated that once the digital product is transmitted in step 1210 to a viewer, which may be any standard viewer appropriate to the content of the digital product, the output stream is no longer secured by the mechanisms built into the overall system as described herein. Accordingly, a step 1208 is optionally inserted between steps 1206 and 1210 of Fig. 12. In an embodiment which includes step 1208, a fingerprint and/or a watermark is (are) inserted into the digital output stream prior to or while it is being provided to the viewer. Watermarking is a technique using a visible identifier that will let the user know that he or she has been associated with this particular instance of the content. It acts primarily as a deterrent. Fingerprinting embeds and hides codes into the output stream itself that are retrievable only by the author or by the licensing authority. Such codes uniquely associate the particular copy of the digital product with the individual who purchased it. Fingerprinting is used primarily for criminal prosecution and court proceedings.

If fingerprinting is used, preferably the fingerprint is inserted in a manner which does not affect the resulting viewing experience. For example, if the output stream includes CD audio, then the fingerprint can be spread over a large number of the audio samples, either substituting for the low-order bit or modifying the low-order bit in an exclusive OR manner in each sample. Alternatively, to avoid differential cryptanalysis, the data stream can be

transformed into the frequency domain, modified in the frequency domain to insert the fingerprint codes, and then transformed back to the time domain. Also for images, steganographic techniques can be used to insert the fingerprint into the image. Steganography is described in Neil F. Johnson, "Steganography", available at http:// adams.patriot.net/ ~johnson/ html/ neil/ stegdoc/ stegdoc.html, visited March 4, 1997, incorporated by reference herein. The fingerprint to be inserted in the digital output stream should preferably be either the installation ID of the reader system 106, or the reader system signature that was generated at the time of product usage (step 1102). Optionally, the output stream can also be randomly seeded to further hamper differential cryptanalysis. In this manner, if pirated copies of a digital product do begin to appear, the author of the product or the licensing authority should be able to determine the original source of the pirated copies by examining the fingerprint.

Returning to Fig. 11, if the two digests do not match (step 1110), then the reader system 106 has determined that the newly generated reader system signature is not the same as that which was generated in step 602 (Fig. 6) at the time of product purchase. In an aspect of the invention, this determination does not immediately preclude usage of the product by the customer on the reader system 106. Instead, proceeding in Fig. 13, in a step 1302, the reader system 106 prepares a re-validation package. The re-validation package can be the same as set forth above with respect to the product purchase package, except that the customer's payment information can be omitted. In step 1304, the reader system 106 uploads the re-validation

package to the license server 104 at the URL identified in the product broadcast package. The license server's operations in response to receipt of a re-validation package are set forth in Fig. 14.

Referring to Fig. 14, in a step 1402, it is first determined whether the RSS in the re-validation package was based on a component in the reader system 106 having external assurances of substantial uniqueness. If so, then re-validation is considered unsuccessful (step 1404) and this result is returned to the reader system 106. If the RSS in the re-validation package was not based on a component having external assurances of substantial uniqueness, then in step 1406, the license server 104 compares the new RSS from the re-validation package to the RSS previously stored accessibly to the server for the same reader system 106 (as identified by the installation ID specified in the re-validation package). If the difference between the two RSS's exceeds the threshold that was specified by the author in the usage parameters stored on the server 104 for the product ID specified in the re-validation package (step 1408), then, again, re-validation is unsuccessful and such a result is returned to the reader system 106 (step 1404). In different embodiments, the threshold can be specified as a percentage of one or the other RSS, or as a number of component signatures which differ between the two RSS's, or by a number of other different specifications.

If the difference between the two RSS's does not exceed the designated threshold (step 1408), then the re-validation is considered successful. The license server 104 prepares a new product server download package using the same algorithms as set forth above

with respect to Fig. 9, but using the new RSS for encryption instead of the RSS that was used to download the original product server download package upon purchase. The new product server download package is then transmitted back to the reader system 106 with re-validation. Optionally, in order to assist investigation of any potential fraud, in step 1410, the license server 104 also stores the new RSS in conjunction with the installation ID specified in the re-validation package. A history of such ostensible reader system upgrades is maintained on the server 104.

Fig. 15 is a detail of step 1408 (Fig. 14) in which the license server 104 determines whether the difference between the two RSS's exceeds the threshold specified by the author in the usage parameters for the digital product. The flowchart set forth in Fig. 15 represents one embodiment, in which the threshold has been specified as a percentage. In a step 1502, the server 104 calculates the weighted sum of the RSS received in the product re-validation package. In a step 1504, the server 104 makes the same calculation with respect to the RSS previously stored on the server 104. In step 1506, the license server 106 determines whether the difference between the two calculated values exceeds the threshold specified by the author in the usage parameters. If so, then in step 1508, the routine returns to Fig. 14 affirmatively. If not, then in step 1510, the routine returns to Fig. 14 negatively.

Fig. 16 is a detail of step 1408 (Fig. 14) as performed in a second embodiment, in which the upgrade drift percentage is specified as a maximum number of components whose individual component signatures can differ between the two RSS's. In step 1602, the server

104 counts the number of components of the RSS in the re-validation package, which differ from the corresponding components of the RSS previously stored on the server 106 from the original product purchase package. If the count exceeds the predetermined drift threshold, then the routine returns affirmatively (step 1606). If not, then it returns negatively (step 1608).

Returning to the reader system flow as illustrated in Fig. 13, after the reader system 106 uploads the re-validation package to the license server 104, in a step 1306, the reader system 106 receives the re-validation result. If the re-validation was unsuccessful (step 1308), then the reader system displays an error message to the user and requests the customer to call customer service of the licensing authority (step 1310). In this situation, automatic re-validation has failed, and manual re-validation as in step 1310 is necessary. During the call, a customer service representative can determine whether the customer's license should be extended to cover the reader system 106 as it now stands. If automatic re-validation was successful (step 1308), then the reader system returns to step 514 (as indicated by the numeral "5" in the small circle in both Figs. 13 and 5) to store and process the new product server download package in the same manner as it processed the original product server download package received upon purchase.

It can be seen that a secure product distribution mechanism has been described which takes advantage of the benefits of an uncontrolled distribution network, while ensuring that authors and proprietors of digital products are paid an appropriate royalty for their efforts at creativity. In addition, the mechanism ensures that once a customer is licensed to use a

digital product on a particular reader system, that product cannot be used on any other reader system without re-validation. The mechanism allows for a certain amount of upgrade drift within which re-validation can be entirely automated.

The foregoing description of preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obviously, many modifications and variations will be apparent to practitioners skilled in this art. For example, whereas the flowcharts described herein illustrate steps being performed in a particular sequence, it will be appreciated that in many instances the sequence of the steps can be reversed, or the steps can be performed in a pipelined, overlapping manner, or both, without departing from the scope of the invention. The embodiments herein were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

CLAIMS

1. A method for preparing a digital product for controlled distribution using a distribution network, comprising the steps of:

encrypting said product;

separating at least one encrypted fragment from said encrypted product;

transmitting said encrypted product less said at least one encrypted fragment onto said distribution network; and

withholding said at least one encrypted fragment from said distribution network.

2. A method according to claim 1, further comprising the step of transmitting said at least one encrypted fragment to a license server.

3. A method according to claim 2, further comprising the step of transmitting to said license server a decryption key that can be used to decrypt said product.

4. A method according to claim 1, wherein said encrypted product includes a header portion followed by a remainder portion,

and wherein said step of separating at least one encrypted fragment from said encrypted product comprises a step of separating from said encrypted product an encrypted fragment that includes at least part of said header portion.